REMARKS

Claims 1-44 are pending in the application.

Claims 1-44 stand rejected.

Claims 8 and 16 have been canceled.

Claims 17-29 and 36-44 have been amended. The amendments to claims 17-29 and 36-44 are not in response to the rejection under 35 U.S.C. § 103.

Claim Rejections - 35 U.S.C. § 103

Claims 1-44 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Krebs et al. U.S. Patent No. 6,668,369 referred to herein as "*Krebs*") in view of Boxall et al. U.S. Patent No. 6,263,456 referred to herein as "*Boxall*"). Applicants respectfully traverse the rejection.

Independent claims 1, 9, 17, 21, 29, and 36 (the "Independent Claims") recite an application program and a debugger program that both "reside on a server that is remote from [a] workstation". The *Office Action* identifies *Krebs* as teaching elements of the Independent Claims involving the application program. However, in contrast to the Independent Claims, the relevant 'application program' taught by *Krebs* does not "reside on a server that is remote from [a] workstation" but rather resides on a client machine. As subsequently discussed in more detail, because the "application program" taught by *Krebs* is not the same as the application program referenced in the Independent Claims, *Krebs* in view of *Boxall* neither teaches nor suggests the Independent Claims.

Additionally, as subsequently discussed in more detail, the relevant debugger programs of *Krebs* and *Boxall* used to display debugging information also reside on a client system rather than the workstation. Accordingly, *Krebs* in view of *Boxall* neither teach nor suggest relevant remote debugging.

Applicants respectfully submit that *Krebs* in view of *Boxall* neither teaches nor suggests, for example:

> invoking the application program and the debugger program using a user interface provided by the web browser and via a network interface to cause the server to execute the application program and the debugger program;
>
> displaying the user frame of the first web page in the web browser at the workstation, wherein the user frame includes information generated by the application program. Claim 1. (emphasis added).

As previously discussed, Claim 1 clearly states that the application program referenced in the above two elements of Claim 1 resides "on a server that is remote from the workstation." **However, the code being debugged by *Krebs* is code generated by a client.** *Krebs* states that, "HTML coding generated dynamically, **i.e., at a client after the file leaves a server**, is referred to as dynamic HTML ("DHTML"). *Krebs*, col. 1, lines 33-35. This code "generated by a client", rather than an application program residing on a remote workstation, is the code referred to by *Krebs* when discussing the invention of *Krebs*. More specifically, *Krebs* teaches that, "The present invention provides a client-side software debugging tool for viewing dynamic code." *Id.*, col. 2, lines 61-63. "Accordingly, the present invention provides a tool for assisting a programmer in locating errors in DHTML and scripts for generating dynamic code.". *Id.*, lines 55-68. "[T]he present invention permits the programmer to view the dynamic code **generated at the client** by the script." *Id.*, lines 61-63.

Krebs identifies the application generating the code at the client for debugging as a "user's computer's web browser". *Id.*, col. 4, line 31. *Krebs* teaches that, "The web browser then runs the script to generate the dynamic code and stores the dynamic code in internal variables of the browser as known in the prior art, as shown at step 54." *Id.*, lines 34-37.

Therefore, the 'application program" generating code to be debugged as taught by *Krebs* in view of *Boxall* does not reside on a server remote from a workstation. Thus, the application program referenced in *Krebs* generates information at the client and, thus, *Krebs* in view of *Boxall* does not teach or suggest:

> invoking the application program and the debugger program using a user interface provided by the web browser and via a network interface to cause the server to execute the application program and the debugger program;

displaying the user frame of the first web page in the web browser at the workstation, wherein the user frame includes information generated by the application program. Claim 1. (emphasis added).

Additionally, *Krebs* teaches that the debugging tool used in *Krebs* resides on a client and not "on a server that is remote from the workstation." Independent Claims. Specifically, *Krebs* teaches that, "The user first executes the software debugging tool, as shown at step 62. This may be achieved in a variety of ways discussed above. For example, if the debugging tool is integrated in the browser, the user may simply select an appropriate menu option." *Krebs*, col. 4, lines 41-45.

Boxall teaches that a debugger tool is located on the client and on the server. However, at least visualization of the debugging operation is accomplished through the client side debugger tool with its own user interface rather than "displaying a debug view option in the web browser at the workstation for generating a second web page having a debug frame of the application program" as required by Claim 1. More specifically, *Boxall* teaches:

> As shown in FIG. 2, the debugger UI 19 comprises a debug daemon 21 and a debug user interface 23. The debug user interface 23 comprises a conventional debugger user interface which displays debug, i.e. trace, information to the program developer and also preferably provides the capability to change variable values, e.g. registers and memory. … The debug daemon 21 is a program which runs in the background on the client machine 3a (or the machine on which the debugger UI 19 is installed). The debug daemon 21 provides an interface between the debugger engine 13 and the debug user interface 23. The debug daemon 21 allows the debugger engine 13 to remotely start the debugger UI 19 on the client machine 3a. *Boxall*, col. 4, lines 10-15 and 21-26.

Thus, *Krebs* in view of *Boxall* neither teaches nor suggests:

debugging an application program from a workstation, wherein the application program and a debugger program reside on a server that is remote from the workstation

invoking the application program and the debugger program using a user interface provided by the web browser and via a network interface to cause the server to execute the application program and the debugger program;

displaying the user frame of the first web page in the web browser at the workstation, wherein the user frame includes information generated by the application program;

receiving the second web page from the server for displaying the debug frame in the web browser at the workstation when the debug view option is selected. **(Claim 1)**.

For at least similar reasons, *Krebs* in view of *Boxall* neither teaches nor suggests:

(Claim 9):

debugging an application program from a workstation, wherein the application program and a debugger program reside on a server that is remote from the workstation
executing the application program and the debugger program on the server when the application program is invoked from the workstation;
generating information for a second web page, wherein the second web page comprises a debug frame when a debug view option is selected from the workstation, wherein the debug frame includes information about components of the application program; and
transmitting the second web page to the workstation.

(Claim 17):

debugging an application program from a workstation, wherein the application program and a debugger program reside on a server that is remote from the workstation
invoking the application program and the debugger program using a user interface provided by the web browser and to cause the server to execute the application program and the debugger program;
presenting the user frame of the first web page in the web browser at the workstation, wherein the user frame includes information generated by the application program;
receiving the second web page from the server for displaying the debug frame in the web browser at the workstation when the debug view option is selected.

(Claim 21):

debugging of an application program from a workstation, wherein the application program and a debugger program reside on a server that is remote from the workstation
executing the application program and the debugger program on the server when the application program is invoked from the workstation;
generating information for a first web page, wherein the first web page comprises a user frame that includes information generated by the application program; and
generating information for a second web page, wherein the second web page comprises a debug frame when a debug view option is selected from the workstation, wherein the debug frame includes information about components of the application program.

(Claim 29)

debugging an application program from a workstation, wherein the application program and a debugger program reside on a server that is remote from the workstation

interact with a web page displayed by the web browser to allow a user to invoke the application program and the debugger program from the workstation to cause the server to execute the application program and the debugger program;

present a first web page in the web browser, wherein the first web page comprises a user frame that includes information generated by the application program;

present a debug view option to generate a second web page having a debug frame of the application program; and

present the debug frame of the second web page when the debug view option is selected, wherein the debug frame includes information about one or more components of the application program.

(Claim 36):

debugging of an application program from a workstation, wherein the application program and a debugger program reside on the server that is remote from the workstation …;

executing the application program and the debugger program on the server when the application program is invoked from the workstation;

generating information for a first web page, wherein the first web page comprises a user frame that includes information generated by the application program; and

a debugger program operable to generate information for a second web page, wherein the second web page comprises a debug frame when a debug view option is selected from the workstation, wherein the debug frame includes information about components of the application program.

For at least the foregoing reasons, Applicants respectfully request withdrawal of the Independent claims and claims directly or indirectly dependent thereon.

Claim Rejections - 35 U.S.C. § 101

I.

Claims 17, 21, and 36 stand rejected under 35 U.S.C. § 101 because all the "means for" elements can be software.

Claim 17 has been amended to include "a processor and a memory coupled to the processor, wherein the workstation is configured for: …".

Claim 21 is now directed to a tangible, computer readable medium.

Claim 36 is a server that "includes a processor and a memory coupled to the processor, wherein the server is configured for: …". Thus, claims 17, 21, and 36 are not directed to software *per se*.

Applicants respectfully request withdrawal of the rejection.

II.

Claims 8 and 16 stand rejected under 35 U.S.C. § 101 because claims 8 and 16 are directed to a signal.

Claims 8 and 16 have been canceled.

Applicants respectfully request withdrawal of the rejection.

Claim Rejections - 35 U.S.C. § 112

Claims 17 and 21 stand rejected under 35 U.S.C. § 112, second paragraph for reciting both an "apparatus" and a "method".

Claim 17 has been amended to delete "apparatus" and "method" and recite a "workstation".

Claim 21 has been amended to delete "apparatus" and "method" and recite a "tangible, computer readable medium".

Applicants respectfully request withdrawal of the rejection.

CONCLUSION

Applicant respectfully submits that all pending claims are in condition for allowance. Accordingly, Applicant requests that a Notice of Allowance be issued. Nonetheless, should any issues remain that might be subject to resolution through a telephone interview, the Examiner is requested to telephone the undersigned at 512-338-9100.

CERTIFICATE OF TRANSMISSION

I hereby certify that on January 21, 2010 this correspondence is being transmitted via the U.S. Patent & Trademark Office's electronic filing system.

/Kent B. Chambers/

Respectfully submitted,

/Kent B. Chambers/

Kent B. Chambers
Attorney for Applicant(s)
Reg. No. 38,839